

Access Control List (ACL)

The Access Control List (ACL) is also called the Object Trustees property. Whenever you make a trustee assignment, the trustee is added as a value to the Object Trustees (ACL) property of the target.

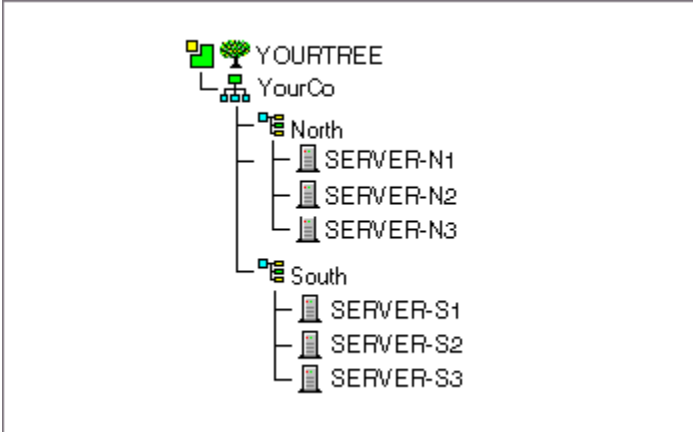
This property has strong implications for network security for the following reasons:

- Anyone who has the Supervisor or Write right to the Object Trustees (ACL) property of an object can determine who is a trustee of that object.
- Any users with the Add Self right to the Object Trustees (ACL) property of an object can change their own rights to that object. For example, they can grant themselves the Supervisor right.

For these reasons, you should be careful giving Add Self rights to all properties of a container object. That assignment makes it possible for the trustee to become Supervisor of that container, all objects in it, and all objects in containers beneath it.

Accommodate Slow or Unreliable WAN Links

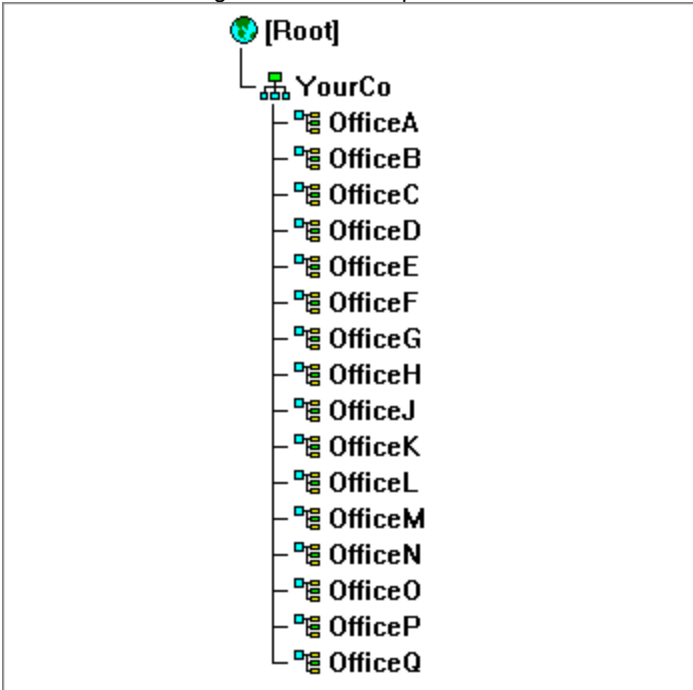
Slow or unreliable WAN links affect your NDS* tree design. You should create containers for each site separated by such links, placing each NetWare* Server object in its local container.



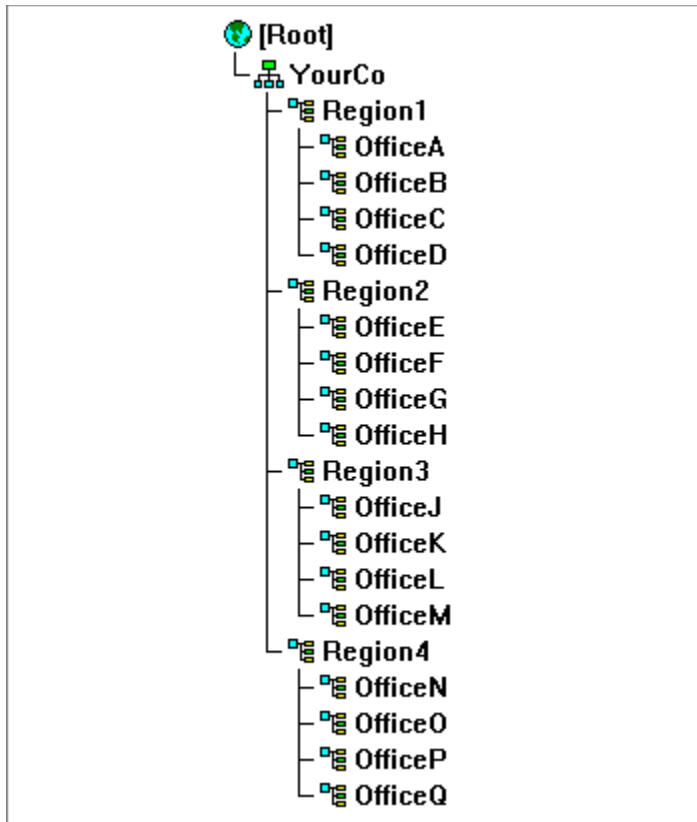
The advantage of this level of site containers is partitioning. Objects representing local resources can be kept on a local partition, eliminating synchronization traffic over the WAN. Another benefit is that resources and volumes are kept in a branch of the tree in common with the users that access them. This simplifies administration of NDS rights and login scripts.

Avoiding a Flat Tree

In organizations with numerous branch offices, you might naturally create a flat tree, with numerous Organizational Units under the Organization. A sample flat tree is shown below.



A flat tree produces inefficiencies that have to be overcome by NDS processes. These inefficiencies can lead to poor performance and a less user-friendly environment for users and administrators. To avoid the inefficiencies of a flat tree, create intermediate containers that subdivide the tree, usually into regions, as shown below.



You may need to further subdivide the tree to restrict partitions to about 1,000 objects. After designing of the upper levels of the tree, plan workgroup containers and leaf objects.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Alias

You can create an Alias object that points to another object in the NDS* tree. Alias objects are most commonly used to give users a local name for an object that lies outside their container.

When you rename a container, you have the option of creating an alias in the former container's place that points to the new name. Workstations and login script commands that reference objects in the container can still access the objects without having the container name updated.

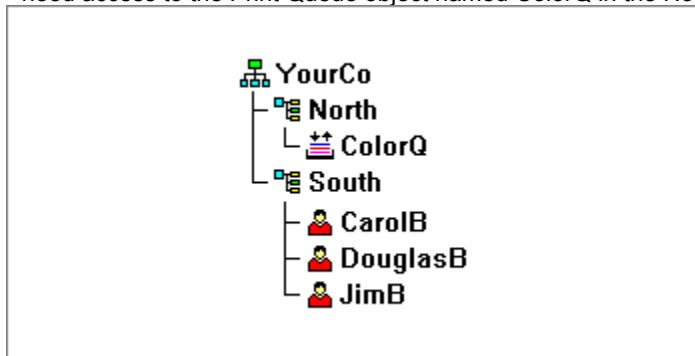
What an Alias Object Represents

An Alias object represents another object, which can be a container, User, or any other object in the tree. An Alias object does not carry trustee rights of its own. Any trustee authority you grant to the Alias object applies to the object it represents. The Alias can be a target of a trustee assignment, though.

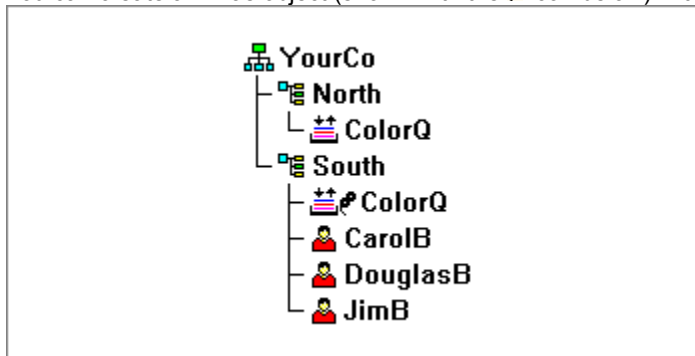
Usage

Create an Alias object to make name resolution easier. Since object naming is simplest for objects in the current context, you should create Alias objects there that point to any resources outside the current context.

For example, suppose users log in and establish a current context in the South container as shown below, but need access to the Print Queue object named ColorQ in the North container.



You can create an Alias object (shown with the  icon below) in the South container.



The Alias points to the original ColorQ object, so setting up printing for the users involves a "local" object.

Important Properties

Alias objects have an Aliased Object property, which associates the Alias with the original object.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Backlink

Backlink verifies external references, which are pointers to Directory objects that are not stored in the replicas on a server.

The backlink process runs two hours after the local database is opened and then every thirteen hours.

Combination of SAP and Time Source List

You can use both SAP and time source lists on the same network. This generates more traffic than using only time source lists but gives you fault tolerance if the servers in the list are not available. You create a combination arrangement using TIMESYNC.CFG commands like the following:

```
# Configuration Parameters from server YOURSERVER

Configured Sources = ON
Service Advertising =  ON

# Configured time source list from server YOURSERVER

Time Source = SERVER1
Time Source = SERVER2
```

The time source list that is stored on a server always takes precedence over the SAP information received by the server. If a server does not have a custom configuration list or if all servers on the list are down, SAP information is used for time synchronization.

Connection Management

Servers in a replica ring require a highly secure connection for transferring NetWare Core Protocol* (NCP*) packets. These secure connections, called virtual client connections, are established by the connection management process.

If a server-to-server connection is lost, the connection management process establishes the connection again when needed.

The connection management process may also need to establish a virtual client connection for schema synchronization or backlink processes. Time synchronization might also require such a connection, depending on the configuration of time services.

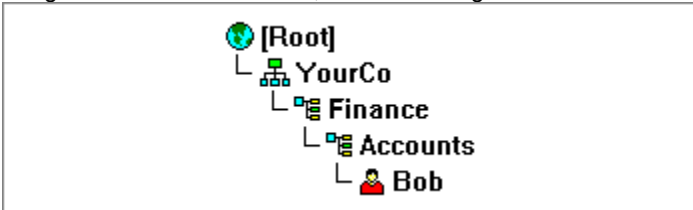
* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Context and Naming

The context of an object is its position in the NDS* tree.

In NWAdmin, the position of the object is shown in the graphic layout.

You can see in the picture below, for instance, that User Bob is in Organizational Unit Accounts, which is in Organizational Unit Finance, which is in Organization YourCo.



Sometimes, however, you need to express the context of an object in an NDS utility. For example, you could be setting up Bob's workstation and need to supply a name context, as shown below.

Client 32 | Login | Default Capture | Advanced Settings

Preferred server: AppServer

Preferred tree: YourTree

Name context: Accounts.Finance>YourCo

First network drive: F

The context is specified as a list of containers separated by periods, between the object in question and the [Root]. In the example above, the User object Bob is in the container Accounts, which is in the container Finance, which is in the container YourCo. (You don't list the container [Root].)

Notice that if you say the context, replacing the periods with the words "which is in," the context seems logical.

Complete Name

The complete name of an object is its object name with the context appended. For example, the complete name of User object Bob is Bob.Accounts.Finance>YourCo.

Typeful Name

Sometimes typeful names are displayed in NDS utilities. Typeful names include the object type abbreviations from the list below:

Object Class	Type	Abbreviation
All leaf object classes	Common Name	CN
Organization	Organization	O
Organizational Unit	Organizational Unit	OU
Country	Country	C

In creating a typeful name, NDS uses the type abbreviation, an equal sign, and the object's name. For instance, Bob's partial typeful name is CN=Bob. Bob's complete typeful name is CN=Bob.OU=Accounts.OU=Finance.O=YourCo. You can use typeful names interchangeably with typeless names in NDS utilities.

Name Resolution

The process NDS uses to find an object's location in the Directory tree is called name resolution. When you use object names in NDS utilities, NDS resolves the names relative to either the current context or the [Root].

Current (Workstation) Context

Workstations have a context set when the networking software runs. For instance, Bob's workstation would be set

to the current context as follows:

```
Accounts.Finance.YourCo
```

Current context is a key to understanding the use of leading periods, relative naming, and trailing periods.

Leading Period

Use a leading period to resolve the name from [Root], no matter where the current context has been previously set. In the example below, the leading period tells the CX (Change Context) utility to resolve the name relative to the [Root].

```
CX .Finance.YourCo
```

NDS interprets the command as "Change context to the Finance container, which is in the YourCo container, resolved from [Root]."

The workstation's current context changes to the Finance container, which is in the YourCo container.

Relative Naming

Relative naming means that names are resolved relative to the workstation's current context, rather than [Root]. Relative naming never involves a leading period, since a leading period indicates resolution from [Root].

Suppose a workstation's current context is set to Finance.

 The relative object name of Bob is

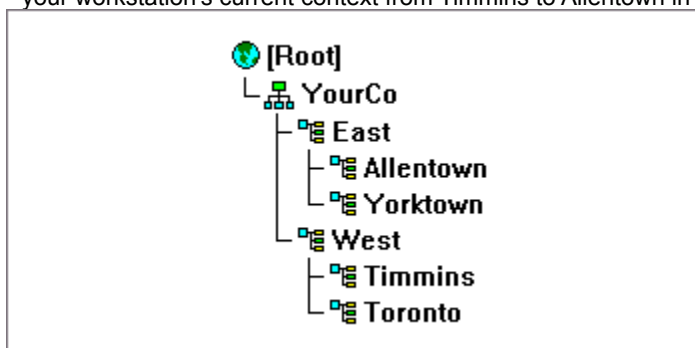
```
Bob.Accounts
```

NDS interprets the name as "Bob, which is in Accounts, resolved from the current context, which is Finance."

Trailing Periods

Trailing periods can only be used in relative naming. Therefore, you can't use both a leading period and a trailing period. A trailing period changes the container from which NDS resolves the name.

Each trailing period changes the resolution point one container toward the [Root]. Suppose you want to change your workstation's current context from Timmins to Allentown in the example below.



The proper CX command uses relative naming with trailing periods:

```
CX Allentown.East..
```

NDS interprets the command as "Change the context to Allentown, which is in East, resolved from two containers up the tree from the current context."

Similarly, if Bob is in the Allentown container and your workstation's current context is Timmins, Bob's relative name is

```
Bob.Allentown.East..
```

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Country

You can create Country objects directly under [Root] using NWAdmin. Country objects are optional and only required for connection to certain X.500 global directories.

What a Country Object Represents

The Country object represents the political identity of its branch of the tree.

Usage

Most administrators do not create a Country object, even if the network spans countries, since the Country object only adds an unnecessary level to the NDS* tree. You can create one or many Country objects under [Root], depending on the multinational nature of your network. Country objects can only contain Organization objects.

If you do not create a Country object and find that you need one later, you can always modify the tree to add one.

Important Properties

The Country object has a two-letter Name property.

Name

Country objects are named with a standard two-letter code such as US, UK, or DE. For a list of standard abbreviations, see also [Naming Countries](#).

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Custom Time Synchronization Configuration

Customizing time synchronization requires a coordinated scheme of communication methods and server types.

Determining a Time Synchronization Communication Method

By default, Primary, Reference, and Single Reference time servers use the Service Advertising Protocol (SAP) to

- Announce their presence on the network
- Determine which other servers to poll in order to negotiate the network time

Secondary time servers use SAP broadcasts from other time server types to choose a time server to synchronize with.

An advantage of the SAP method is that it allows for quick installation without regard to the network layout. It also allows automatic reconfiguration if operating modes are changed or if new servers are added to the network.

The SAP method, however, generates more network traffic than a customized method.

The SAP method can also take longer to bring large network environments into synchronization, particularly if test servers are configured as time sources.

Customized time server communication involves a time source list or a combination of SAP and time source list

Determining Time Server Types

In creating a customized master plan of time server types, you should consider your network topology and your applications' requirements for accurate time. Accurate time stamps are best achieved using external time sources. To maintain fault tolerance and reduce network traffic, consider planning a time provider group. You can also combine the two schemes as your situation demands.

Default Rights for a New NetWare Server

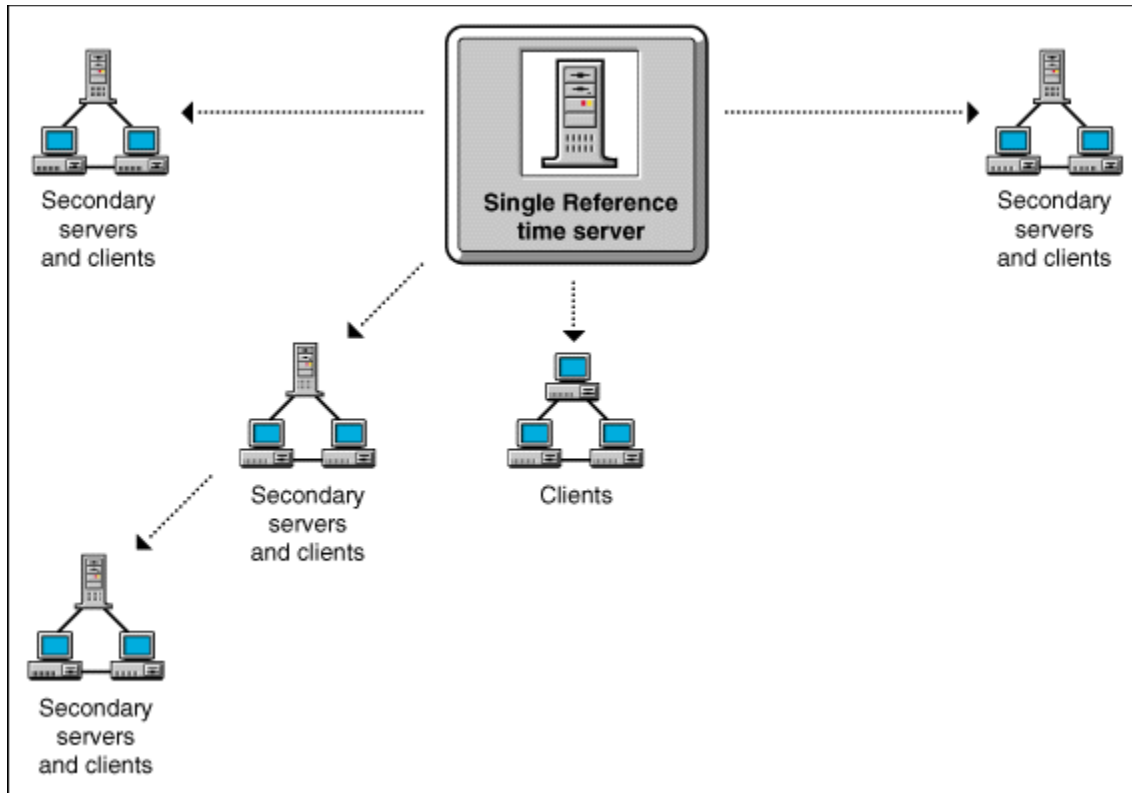
When you install a new NetWare* Server object into an NDS tree, the following NDS trustee assignments are made:

Default Trustees	Default Rights
Admin (first <u>NDS server</u> in the tree)	Supervisor object right to [Root]
[<u>Public</u>] (first NDS server in the tree)	Browse object right to [Root]
NetWare Server	Admin has the Supervisor object right to the NetWare Server object, which means that Admin also has the Supervisor right to the root directory of the file system of any NetWare volumes on the server.
Volumes (if created)	[Root] has the Read property right to the Host Server Name and Host Resource properties on all Volume objects. This gives all objects access to the physical volume name and physical server name. Admin has the Supervisor right to the root directory of the file systems on the volume. For volume SYS, the container object has Read and File Scan rights to the volume's \PUBLIC directory. This allows User objects under the container to access NetWare utilities in \PUBLIC.
User	If home directories are automatically created for users, they have the Supervisor right to those directories.

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

Default Time Synchronization Configuration

By default, the first server on the network is a Single Reference time server and subsequent servers are Secondary time servers (see the figure below). This configuration works well for most networks.



A limitation of a Single Reference time server configuration is the lack of fault tolerance if the time server loses its connection for an extended period of time. This can cause Secondary servers to fall out of synchronization. However, if the Single Reference time server loses its connection, you can designate one of the Secondary time servers as a temporary Single Reference time server until the original is reconnected.

Important: If you use a Single Reference time server, avoid using Reference time servers in the same tree because the time references might conflict.

If your network has any of the following, you should consider a custom time synchronization configuration.

- 🐞 WAN links between servers
- 🐞 Limited bandwidth for network synchronization traffic
- 🐞 Requirements for highly accurate time

Delegated Administration

NDS allows you to delegate administration of a branch of the NDS* tree, revoking your own management rights to that branch. One reason for this approach might be that special security requirements require a different administrator with complete control over that branch.

Delegated administration is accomplished by

1. Granting the Supervisor object right to a container.
2. Creating an IRF at that container which filters the Supervisor and any other rights you want blocked.

Caution: If you delegate administration to a User object and that object is subsequently deleted, there will be no objects with NDS rights to manage that branch.

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

Directory Map

A Directory Map object is a pointer to a path in the NetWare* file system. It allows you to make simpler references to directories.

If your network has no NetWare volumes, you cannot create Directory Map objects.

What a Directory Map Object Represents

A Directory Map object represents a directory on a NetWare volume. (An Alias, on the other hand, represents an NDS* object.)

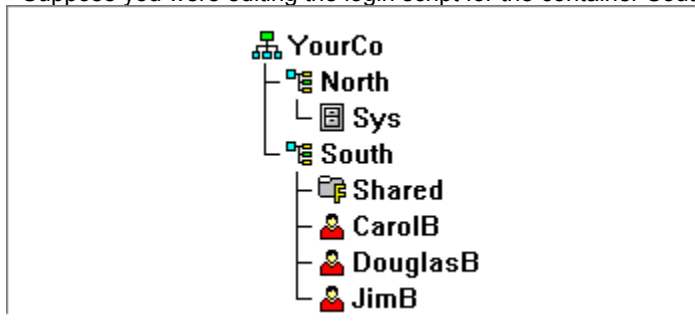
Usage

Create a Directory Map object to make drive mapping simpler, particularly in login scripts. Using a Directory Map object allows you to reduce complex file system paths to a single name.

Also, when you change the location of a file, you don't need to change login scripts and batch files to reference the new location. You only need to edit the Directory Map object.


Example

Suppose you were editing the login script for the container South, shown below.



A command mapping drives to the SHARED directory on volume SYS would look like the following:

```
MAP N:=SYS.North.:Shared
```

If you created the Shared Directory Map object (represented by the  icon), the map command would be much simpler:

```
MAP N:=Shared
```

Important Properties

The Directory Map object has Name, Volume, and Path properties.

Name

The Name property identifies the object in the Directory (for example, Shared), and is used in MAP commands.

Volume

The Volume property contains the name of the Volume object that the Directory Map object references, such as Sys.North.YourCo.

Path

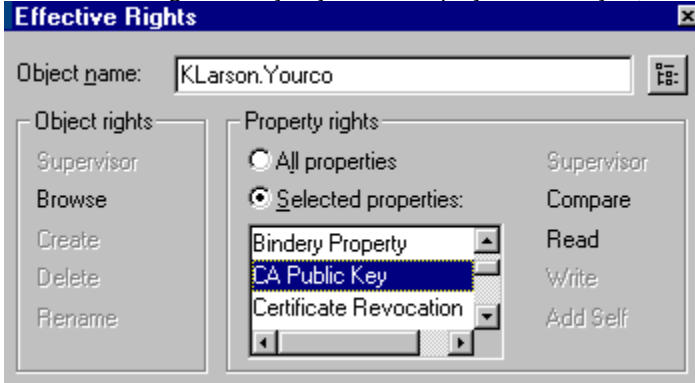
The Path property specifies the directory as a path from the root of the volume, such as PUBLIC\WIN95\NLS\ENGLISH.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Effective Rights

Users can receive rights in a number of ways, such as explicit trustee assignments, inheritance, and security equivalence. Rights can also be limited by IRFs and changed or revoked by subsequent trustee assignments. The net result of all these actions--the rights a user can employ--are called effective rights.

A user's effective rights to an NDS* object are calculated each time the user attempts an action. NWAdmin checks the effective rights of any object and displays them for you, as shown below.



The effective rights displayed for User KLarson are

- ☛ Browse object right
- ☛ Compare and Read right to the CA Public Key property. (To view rights to other properties, select the property from the Selected Properties list.)

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

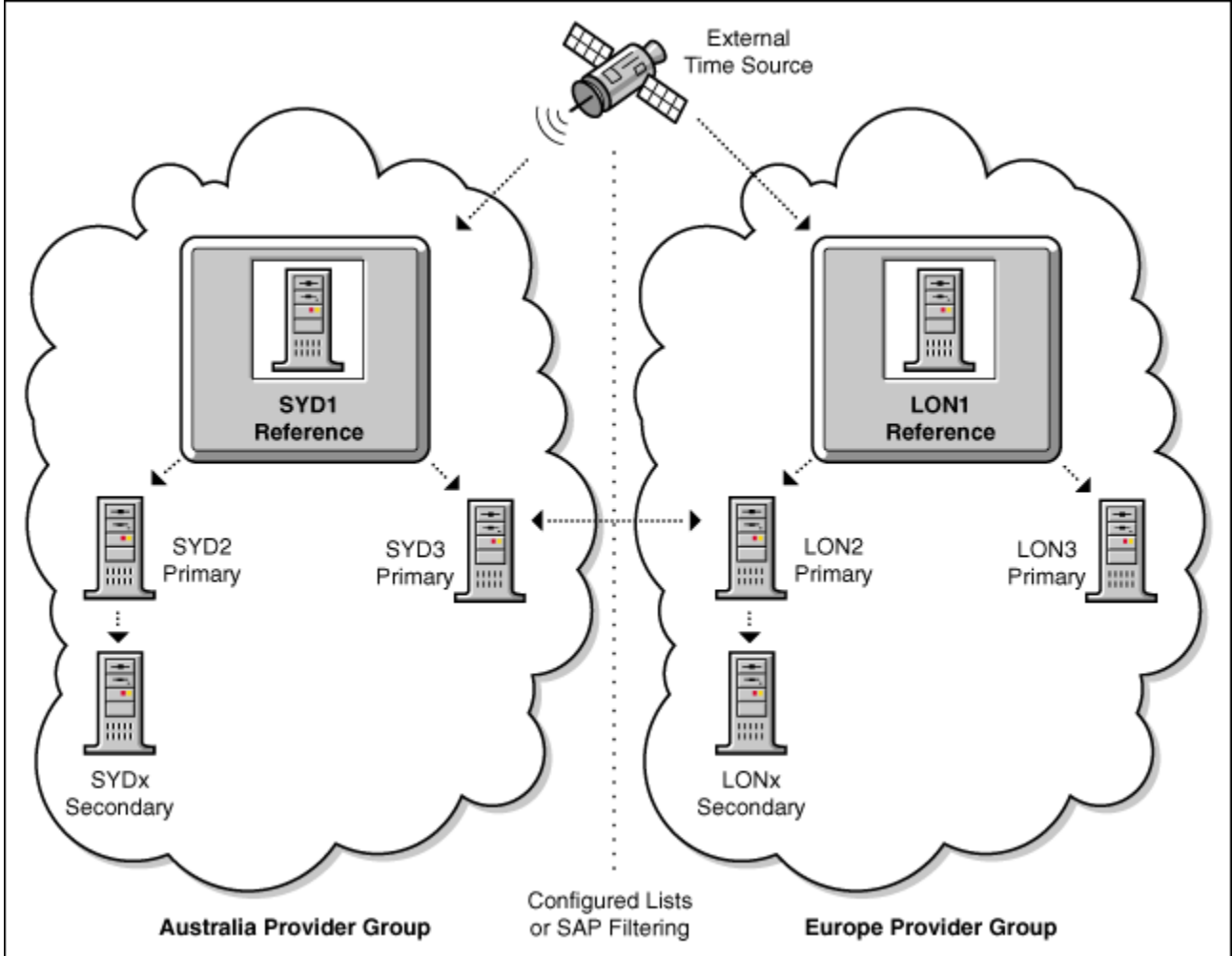
External Time Sources

Synchronized time means servers agree on the *same* time, not necessarily that it is the *correct* time.

There are cases when a network needs to have the correct time across all servers. This can be critical for run time applications, for example. In these cases the best option is to have the tree's reference server (which can be a Single Reference or Reference server) synchronize to an atomic clock rather than relying on its own hardware clock.

There are a few third party applications available to do this, usually requiring an NLM to run on the reference server and dial out via modem or listen to a particular radio band. The NLM takes control of the time synchronization process and uses an external atomic clock to set the network's time.

If there are multiple trees on a LAN or WAN, all trees can use the external time source, as shown below.



This is best accomplished by setting up the main server (the one running the NLM) as a Reference server, then having at least one other Primary server in that "main" tree. Then each additional tree on the LAN can be set up with its main server as a Primary server and set to poll the Reference server of the main tree in its configured sources list.

Then all Secondary servers in all trees could get their time from their Primary server (or Reference for servers in the main tree). In order for that to work, however, Directory Tree Mode needs to be turned off for that primary server.

It would probably be a good idea with the previous scenario to also have at least one other primary in each tree in case there is ever a link down in the network for an extended period of time. Another possible scenario that would avoid this case is to have both sides of the link dial up to an external source independently, which would still give the precise time to both sides and not depend on the link being continuously up.

The only change needed in the TIMESYNC.CFG file for external time sources to work is that the Timesync Hardware Clock option should be set to OFF. This will prevent NetWare* from trying to read its own hardware

clock and possibly causing a conflict.

There may also be an issue of how external sources address daylight savings time. This should be checked with the third party vendor to ensure proper configuration.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Group

You can create Group objects in NWAdmin to help you manage sets of User objects.



What a Group Object Represents

A Group object represents a set of User objects.

Usage

While container objects let you manage all User objects in that container, Group objects are for subsets within a container or in multiple containers.

Group objects have two main purposes:

-  They allow you to grant rights to a number of User objects at once.
-  They allow you to specify login script commands using "IF MEMBER OF..." syntax.

Important Properties

The most important properties of the Group object are Members and Rights to Files and Directories. For a complete list of properties, select a Group object in NWAdmin, then choose Object > Details. To display a description for each page of properties, click Help.

Members

This property lists all User objects in the group. Rights assignments made to the Group object apply to all members of that group.

Rights to Files and Directories

This property lists all trustee assignments made for this Group to the NetWare* file system.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Heartbeat

Heartbeat ensures that Directory objects are consistent among all replicas of a partition. This means that any server with a copy of a partition must communicate with the other servers to check the consistency.

Heartbeat runs by default once every 30 minutes on every server that contains a replica of a partition.

How TIMESYNC Works

TIMESYNC determines if its internal clock is synchronized within a specific time radius of the value received from the time provider. If the synchronization radius is within the set amount of time, the server sends a time synchronization flag to indicate that synchronization is established.

The synchronization radius is by default 2000 milliseconds, or roughly 2 seconds. You can change the synchronization radius with server SET commands.

Since servers can exist in different time zones, the server's internal time is adjusted for time zone offsets and daylight saving time offsets. The resulting time is called Universal Coordinated Time (UTC), which is commonly called Greenwich Mean Time. Servers exchange time synchronization information in UTC time.

TIMESYNC reads time zone and daylight saving time adjustments from parameters set in the AUTOEXEC.NCF file. Here is a sample of those parameters:

```
set Time Zone = MST7MDT
set Daylight Savings Time Offset = 1:00:00
set Start Of Daylight Savings Time = (APRIL SUNDAY FIRST 2:00:00 AM)
set End Of Daylight Savings Time = (OCTOBER SUNDAY LAST 2:00:00 AM)
```

Another command in the AUTOEXEC.NCF file (or optionally in the TIMESYNC.CFG file) sets the time server type:

```
set Default Time Server Type = SECONDARY
```

The time server type determines which servers adjust their clocks to maintain a common time source. Time synchronization provided in IntranetWare* and NetWare* supports four time server types:

Primary

Secondary

Single Reference

Reference

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

If Your Network Needs a Bindery

Many applications, such as print servers and backup software, were written for NetWare* versions previous to NetWare 4*. These applications used the NetWare bindery instead of NDS* for network access and object manipulation.

The bindery is a flat database of objects such as Users, Groups, and Volumes known to a given server. The bindery is server-specific and server-centric.

Older NetWare client software (such as the NETX bindery shell) used a bindery login procedure in which a user logged in to a specific server only. Access to multiple servers required multiple logins using multiple user accounts.

NDS allows applications written for a bindery to function using bindery services. Bindery services allows you to set a context or a number of contexts as a server's virtual bindery. The context you set is called a server's bindery context.

Following are some important facts about bindery services:

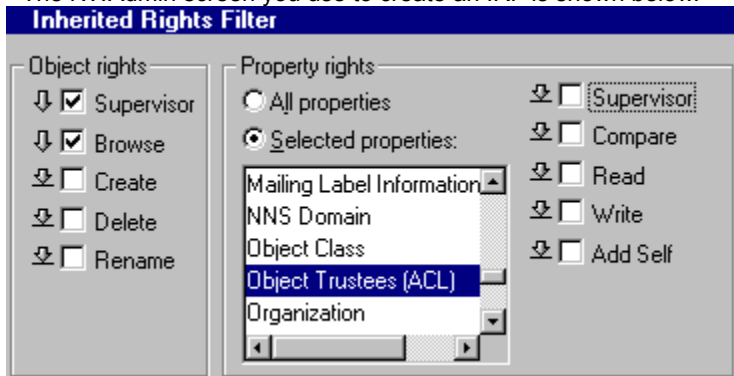
- ☛ To use bindery services, you must set a bindery context for the server.
- ☛ Not all NDS objects map to bindery objects. Many NDS objects, such as Alias objects, do not have a bindery equivalent.
- ☛ Most bindery applications have been upgraded to work with NDS. Check with your application vendor to get the newest version.
- ☛ Each server with a bindery context must hold a master or read/write replica of the partition that includes the bindery context.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Inherited Rights Filter (IRF)

An Inherited Rights Filter (IRF) limits the inheritance of rights. When you create an IRF, you specify the rights which can be inherited from the parent object.

The NWAdmin screen you use to create an IRF is shown below.



Notice the check marks under Object rights next to Supervisor and Browse. These indicate that objects inherit the Supervisor and Browse rights assigned to container targets higher in the tree (but no other rights).

Notice also that under Property rights, no rights are selected to the Object Trustees (ACL) property. This indicates that no assignments to that property will be inherited from containers higher in the tree.

NWAdmin will not allow you to filter the Supervisor object right without granting that right explicitly. Such an IRF would cut off all management of the target object and all objects lower in the tree.

IntranetWare Time Synchronization

All servers that run the NetWare 4* operating system are time servers of some type. They all run the TIMESYNC utility (TIMESYNC.NLM) and exchange information about time to stay synchronized.

Each time-synchronized server performs two fundamental functions. It

- Provides time to any NLM or client workstation that requests it
- Provides status information indicating whether the time is synchronized

For more information, see

- [How TIMESYNC Works](#)
- [Default Time Synchronization Configuration](#)
- [Custom Time Synchronization Configuration](#)

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Limber

Limber ensures that the replica pointer table is updated when a server name or address is changed. Such changes occur when

- The server is rebooted with a new server name or IPX* internal address.
- An address is added for an additional protocol.

When a server is booted, the limber process compares the server's name and IPX address with those stored in the replica pointer table. If either are different, NDS* automatically updates all replica pointer tables that contain a listing of that server.








The limber process also checks that the tree name is correct for each server in a replica ring.

Limber runs by default five minutes after the server boots up and then every three hours.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

List of Objects

The following tables list NDS* object classes. Note that added services can create new object classes in NDS that are not listed below. Also, all classes may not be available on all server operating systems hosting NDS.

Container Object (Abbreviation)	Description
 [Root]	Represents the beginning of your NDS tree.
 Country (C)	Designates the countries where your network resides; also organizes other Directory objects within the country. See also Country .
 Licensed Product (LP)	Created automatically when you install a license certificate or create a metering certificate using Novell Licensing Services (NLS) technology. When an NLS-enabled application is installed, it adds a Licensed Product container object to the NDS tree and a License Certificate leaf object to that container.
 Organization (O)	Helps you organize other objects in the Directory. The Organization object is a level below the Country object (if you use the Country object). See also Organization .
 Organizational Unit (OU)	Helps you to further organize other objects in the Directory. The Organizational Unit object is a level below the Organization object. See also Organizational Unit .
Leaf Object	Description
 AFP Server	Represents an AppleTalk** Filing Protocol server that operates as a node on your NDS network. It usually also acts as a NetWare router to, and the AppleTalk server for, several Macintosh** computers.
 Alias	Points to the actual location of an object in the Directory. Any Directory object located in one


place in the Directory can also appear to be in another place in the Directory by using an Alias. See also [Alias](#).

 Application

Represents a network application. Application objects simplify administrative tasks such as assigning rights, customizing login scripts, and launching applications.

 Computer


Represents a computer on the network.

 Directory Map


Refers to a directory in the file system. See also [Directory Map](#).

 Group

Assigns a name to a list of User objects in the Directory. You can assign rights to the group instead of to each user, then the rights transfer to each user in the group. See also [Group](#).

 License Certificate

Used with Novell Licensing Services (NLS) technology to install product license certificates as objects in the NDS database. License Certificate objects are added to the Licensed Product container when an NLS-aware application is installed.

 NetWare Server

Represents a server running any version of NDS. See also [NetWare Server](#).

 Organizational Role







Defines a position or role within an organization.

 Print Queue

Represents a network print queue.

 Print Server

Represents a network print server.

 Printer	Represents a network printing device.
 Profile	Represents a login script used by a group of users who need to share common login script commands. The users don't have to be in the same container in the NDS tree. See also Profile .
 Template	Represents standard User object properties that can be applied to new User objects.
 User	Represents the people who use your network. See also User .
 Unknown	Represents an NDS object that has been corrupted and can't be identified as belonging to any of the other object classes.
 Volume	Represents a physical volume on the network. See also Volume .

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

NDS Ease of Management

The NDS* Directory allows for easy, powerful, and flexible management of network resources. It also serves as a repository of user information for groupware and other applications. These applications access your NDS Directory through the industry-standard Lightweight Directory Access Protocol (LDAP).

NDS ease-of-management features include a powerful tree structure, an integrated management utility, and single login and authentication.

Powerful Tree Structure

NDS organizes objects in a tree structure, beginning with an object called [Root].

Whether your [NDS servers](#) are running NetWare, UNIX**, or Windows NT**, all resources can be kept in the same NDS tree. You won't need to access a specific server or domain to create objects, grant rights, change passwords, or manage applications.

The hierarchical structure of the NDS tree gives you great management flexibility and power. These benefits are primarily the result of two features: container objects and inheritance.

Container Objects

Container objects allow you to manage other objects in sets, rather than individually. There are three common classes of container objects:



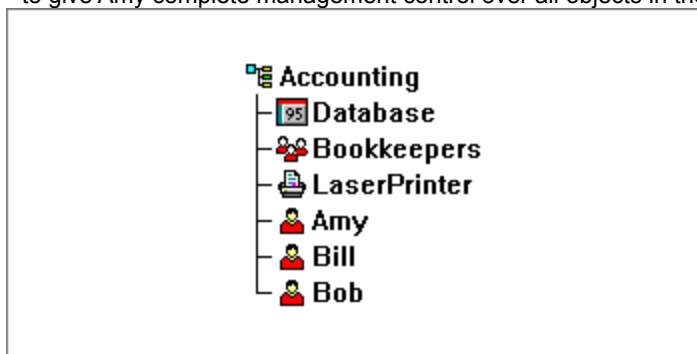
[Root] is the first container object in the tree. It usually contains your company's Organization object.

Organization is normally the first container class under [Root]. The Organization object is typically named after your company. Small companies keep management simple by having all other objects directly under the Organization object.

Organizational Unit objects can be created under the Organization to represent distinct geographical regions, network campuses, or individual departments. You can also create Organizational Units under other Organizational Units to further subdivide the tree.

A fourth class of container object is the Country object, which is seldom used.

You can perform one task on the container object that applies to all objects within the container. Suppose you want to give Amy complete management control over all objects in the Accounting container.



All you need to do is drag the User object named Amy onto the Organizational Unit object named Accounting. Then you select the rights you want Amy to have and click OK. Now Amy has rights to manage the Database application, the Bookkeepers group, the LaserPrinter printer, and the User objects Amy, Bill, and Bob.

For more information on NDS objects, see Object Classes and Properties.

Inheritance

Another powerful feature of NDS is rights inheritance. Inheritance means that rights "flow down" to all containers in the NDS tree. This allows you to grant NDS rights with very few rights assignments. For example, suppose you want to grant management rights to the NDS objects shown below:

You could make any of the following assignments:

If you grant a user rights to Allentown, the user can only manage objects in the Allentown container.

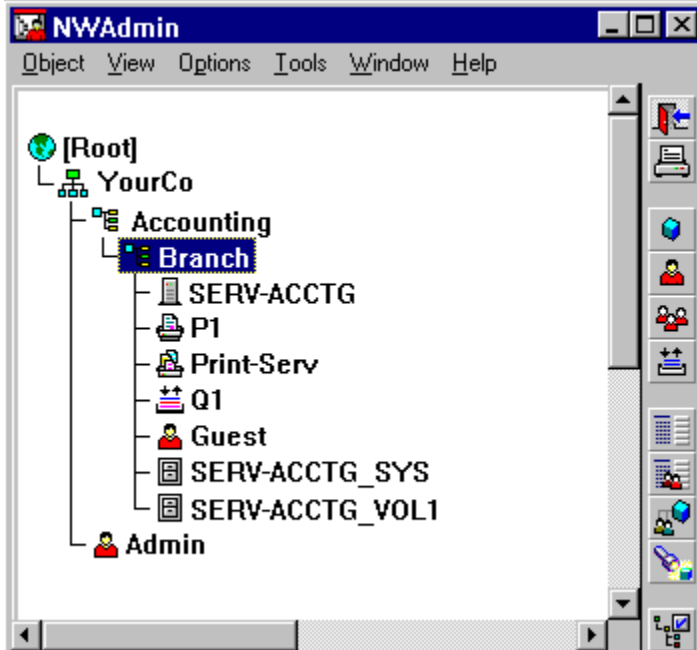
If you grant a user rights to East, the user can manage objects in the East, Allentown, and Yorktown containers.

If you grant a user rights to YourCo, the user can manage any objects in any of the containers shown.

For more information on assigning rights, see NDS Rights.

Integrated Management Utility

NWAdmin (shown below) is the management console for the entire network.



From a workstation running Windows** 3.1x, Windows 95**, or Windows NT**, you can use NWAdmin to perform the following supervisory tasks:

- Create and delete NDS* objects
- Move and rename NDS objects
- Assign rights in the NDS tree and in the NetWare file system
- Set up print services
- Set up licensing services

Note: NWAdmin incorporates all the functions available in Novell's DOS command line utilities FILER, NETADMIN, PARTMGR, and PCONSOLE.

Single Login and Authentication

Users log in to a global directory, so you don't need to manage multiple server or domain accounts for each user, and you don't need to manage trust relationships or pass-through authentication among domains.

A security feature of the Directory is authentication of users. Before a user logs in, a User object must be created in the Directory. The User object has certain properties, such as a name and password.

When the user logs in, NDS checks the password against the one stored in the Directory for that user and grants access if they match.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

NDS Rights

When you create an NDS* tree, the default rights assignments give your network generalized access and security. Some of the default assignments are as follows:

☛ User Admin has the Supervisor right to [Root], giving Admin complete control over the entire Directory. Admin also has the Supervisor right to the NetWare* Server object, giving complete control over any volumes on that server.

☛ [Public] has the Browse right to [Root], giving any user the right to view any objects in the NDS tree.

☛ Objects created through an upgrade process such as a NetWare migration, printing upgrade, or Windows NT** user migration receive NDS trustee assignments appropriate for most situations.

For an explanation of trustees and target objects, see [Trustee Assignments and Targets](#).

NDS Rights Concepts

Following is a list of concepts that will help you understand NDS rights.

[Inheritance](#)

[Object Rights](#)

[Property Rights](#)

[Effective Rights](#)

[Security Equivalence](#)

[Access Control List \(ACL\)](#)

[Inherited Rights Filter \(IRF\)](#)

[Default Rights for a New NetWare Server](#)

[Delegated Administration](#)

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

NDS Server

A server with NDS installed. NDS servers can hold replicas of the NDS database and synchronize those replicas with other NDS servers. NDS servers are represented in the NDS tree by NetWare Server objects.

NetWare Server

A NetWare* Server object is created in the NDS* tree automatically whenever you install NDS on a server. Although the object class is called a *NetWare* Server, it can be an IntranetWare*, UNIX**, or other server running NDS.

You can also create a NetWare Server object to represent a NetWare 2 or NetWare 3* bindery server.

What a NetWare Server Object Represents

The NetWare Server object represents a server running NDS, or a bindery-based (NetWare 2 or NetWare 3) server.

Usage

The NetWare Server object serves as a reference point for replication operations. A NetWare Server object that represents a bindery-based server allows you to manage the server's volumes with NWAdmin.

Important Properties

The NetWare Server object has a Network Address property, among others. For a complete list of properties, select a NetWare Server object in NWAdmin, then choose Object > Details. To display a description for each page of properties, click Help.

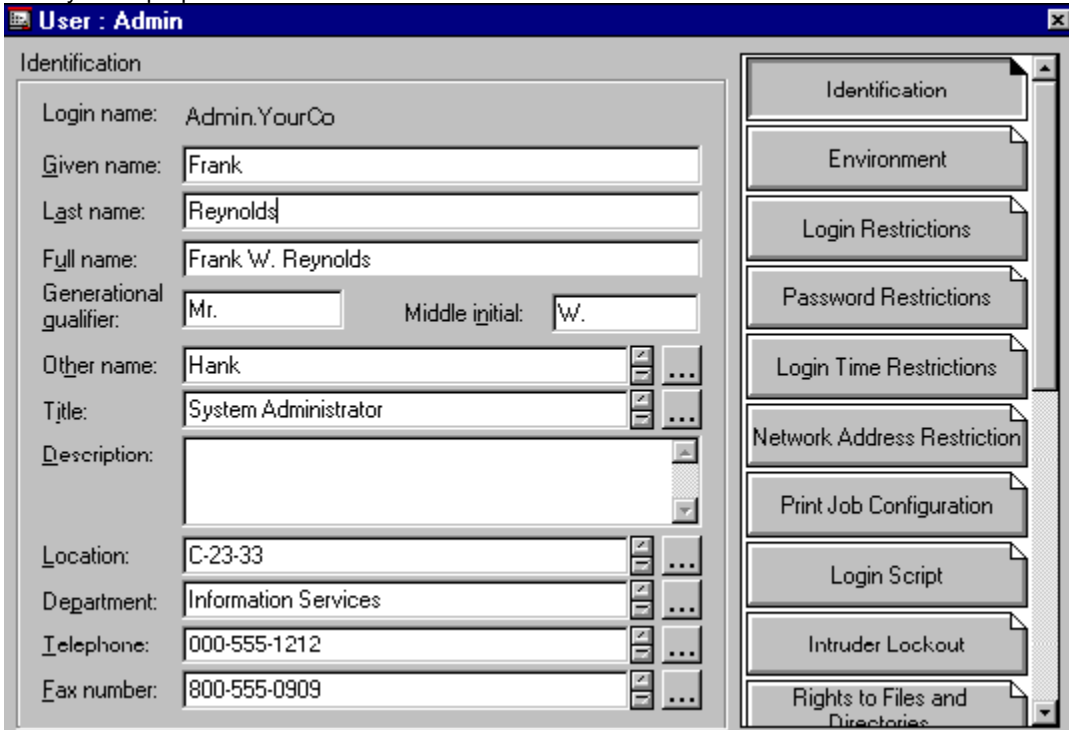
Network Address

The network address property displays the protocol and address number for the server. This is useful for troubleshooting at the packet level.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Object Classes and Properties





Each type of NDS* object is called an object class. For instance, User and Organization are object classes. Each class of object has certain properties. A User object, for example, has Login Name, Password, Last Name, and many other properties.










A set of rules called the schema defines the object classes and properties, along with the rules of containment (what containers can contain which objects). NDS ships with a base schema that you, or the applications you use, can extend.

Container objects contain other objects and are used to divide the tree into branches, while leaf objects represent network resources. The object classes listed below don't constitute an exhaustive list. These are only some often-used classes. For more information, click on an object or its icon below. For classes not shown, see List of Objects.

Container Object Classes

-  [Root]
-  Organization
-  Organizational Unit
-  Country

Leaf Object Classes

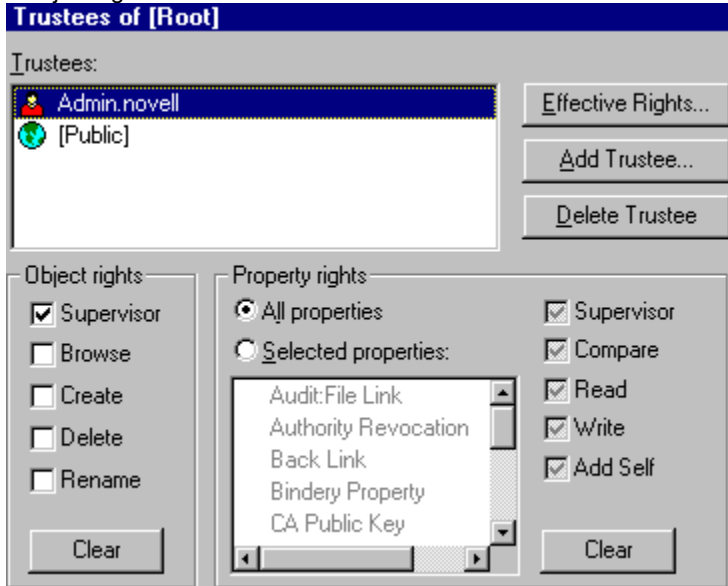
-  NetWare Server
-  Volume
-  User
-  Group
-  Alias
-  Directory Map
-  Profile

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

Object Rights

When you make a trustee assignment, you can grant object rights and/or property rights. Object rights apply to manipulation of the entire object, while property rights only apply to certain object properties.

Object rights are shown in the left-hand side of the NWAdmin Trustees dialog. A description of each right follows.



Supervisor includes all rights to the object and all of its specific properties.

Browse enables the trustee to see the object in the tree. It does not include the right to see an object's specific properties.

Create only applies when the target object is a container. The target in the example above is [Root], shown in the window's title bar. Create allows the trustee to create new objects below the container, and also includes the Browse right.

Delete allows the trustee to delete the target from the Directory.

Rename allows the trustee to change the name of the target.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Organization

An Organization container object is created when you first install NDS* on a server in your network. As the topmost container under [Root], it usually holds Organizational Unit objects and leaf objects.

The User object named Admin is created by default in your first Organization container.

What an Organization Object Represents

Normally the Organization object represents your company, although you can create additional Organization objects under [Root]. This is typically done for networks with distinct geographical districts or for companies with separate NDS trees that have merged.

Usage

The way you use Organization objects in your tree depends on the size and structure of your network. If the network is small, you should keep all leaf objects under one Organization object.

For larger networks, you can create Organizational Unit objects under the Organization to make resources easier to locate and manage. For instance, you could create Organizational Units for each department or division in your company.

For networks with multiple sites, you should create an Organizational Unit for each site under the Organization object. That way, if you have (or will have) enough servers to partition the Directory, you can do so logically along site boundaries.

For easy sharing of company-wide resources, such as printers, volumes, or applications, create corresponding Printer, Volume, or Application objects under the Organization.

Important Properties

The following properties (and many more) are available for the Organization object. Only the Name property is required. For a complete list of properties, select an Organization object in NWAdmin, then choose Object > Details. To display a description for each page of properties, click Help.

Name

Typically, the Name property is the same as your company's name. Of course, you can shorten it for simplicity. For instance, if the name of your company is Your Shoe Company, you might use YourCo.

The Organization name becomes part of the context for all objects created under it.

Login Script

The Login Script property contains commands that are executed by any User objects directly under the Organization. These commands are run when a user logs in.

Applications

The Applications property lists all applications you have set up for this context. Users view and work with applications via the Novell* Application Launcher (NAL) Window and NAL Explorer desktop software.

Applications are associated with an Organization from the Application object's Associations property page. When an application is associated with an Organization object, User objects in that Organization are (by default) presented the application through the NAL Window.

Launcher Configuration

Use the Launcher Configuration properties to specify how users view and work with the Novell Application Launcher (NAL) Window and NAL Explorer desktop software. Use this property to allow users to inherit applications through multiple levels of the NDS tree.

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

Organizational Unit (OU)

You can create Organizational Unit (OU) container objects to subdivide the NDS* tree. Organizational Units are created with NWAdmin under an Organization, Country, or another Organizational Unit object.

Organizational Units can contain other Organizational Units and leaf objects such as User and Application objects.

What an Organizational Unit Object Represents

Normally the Organizational Unit object represents a department, which holds a set of objects that commonly need access to each other. A typical example is a set of Users, along with the Printers, Volumes, and Applications that those Users need.

At the highest level of Organizational Unit objects, each Organizational Unit can represent each site (separated by WAN links) in the network.

Usage

The way you use Organizational Unit objects in your tree depends on the size and structure of your network. If the network is small, you probably don't need any Organizational Units.

For larger networks, you can create Organizational Unit objects under the Organization to make resources easier to locate and manage. For instance, you could create Organizational Units for each department or division in your company. Remember that administration is easiest when you keep User objects together in the Organizational Unit with the resources they use most frequently.

For networks with multiple sites, you should create an Organizational Unit for each site under the Organization object. That way, if you have (or will have) enough servers to partition the Directory, you can do so logically along site boundaries.

Important Properties

The following properties (and many more) are available for the Organizational Unit object. Only the Name property is required. For a complete list of properties, select an Organizational Unit object in NWAdmin, then choose Object > Details. To display a description for each page of properties, click Help.

Name

Typically, the Name property is the same as the department name. Of course, you can shorten it for simplicity. For instance, if the name of your department is Accounts Payable, you can shorten it to AP.

The Organizational Unit name becomes part of the context for all objects created under it.

Login Script

The Login Script property contains commands that are executed by any User objects directly under the Organizational Unit. These commands are run when a user logs in.

Applications

The Applications property lists all applications you have set up for this context. Users view and work with applications via the Novell* Application Launcher (NAL) Window and NAL Explorer desktop software.

Applications are associated with an Organizational Unit from the Application object's Associations property page. When an application is associated with an Organization object, User objects in that Organization are (by default) presented the application through the NAL Window.

Launcher Configuration

Use the Launcher Configuration properties to specify how users view and work with the Novell Application Launcher (NAL) Window and NAL Explorer desktop software. Use this property to allow users to inherit applications through multiple levels of the NDS tree.

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

Partitions: Helping NDS Perform Well

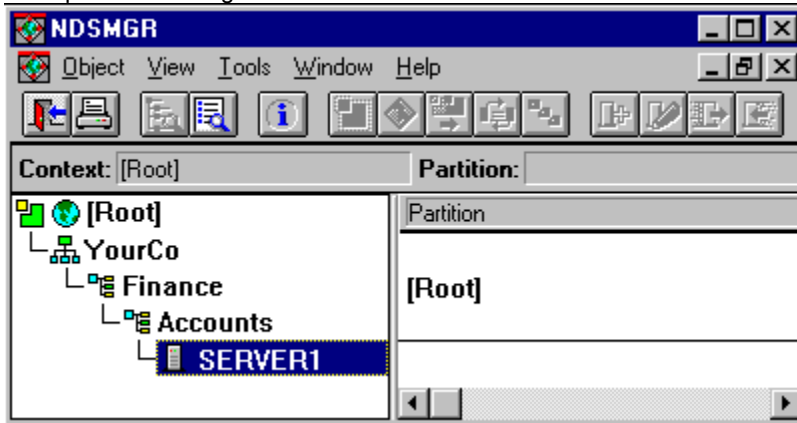
If you have more than 1,000 objects in your NDS* tree, or slow or unreliable WAN links, you should consider partitioning (dividing) the NDS Directory.

Partitioning allows you to take part of the Directory off one server and put it on another server.

A partition is a logical division of the NDS database. A Directory partition forms a distinct unit of data in the NDS tree that stores Directory information.

Each Directory partition consists of a set of container objects, all objects contained in them, and data about those objects. NDS partitions don't include any information about the file system or the directories and files contained there.

Partitioning is done with the NDS Manager utility, which you can access from NWAdmin or run independently. A sample NDS Manager screen is shown below.



Partitions are identified in NDS Manager by the partition icon.

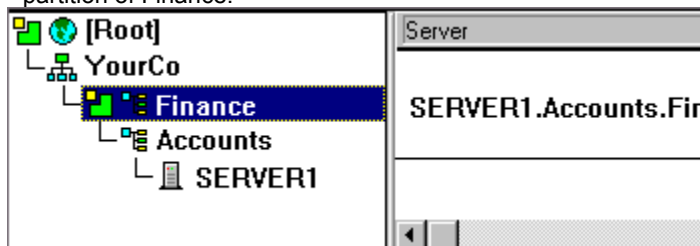
In the example, the partition icon is next to [Root]. This means [Root] is the topmost container in the partition. No partitions are shown by any other containers, so the [Root] partition is the only one.

This is the default partitioning for NDS, keeping the entire Directory together in one partition.

Notice in the screen that server SERVER1 is highlighted. When you highlight a server in NDS Manager, any replicas held on that server are shown on the right. In this case, SERVER1 holds a replica of the [Root] partition. See also [Replicas: Keeping NDS Running](#).

Creating a New Partition

Partitions are named by their topmost container. In the example below there are two partitions, named [Root] and Finance. Finance is called a child partition of [Root], since it was split off from [Root]. [Root] is called the parent partition of Finance.



You might create such a partition because the Directory has so many objects that the server is overwhelmed and access to NDS is slow. Creating the new partition allows you to split the database and pass the objects in that branch to a different server.

Distributing Replicas for Performance

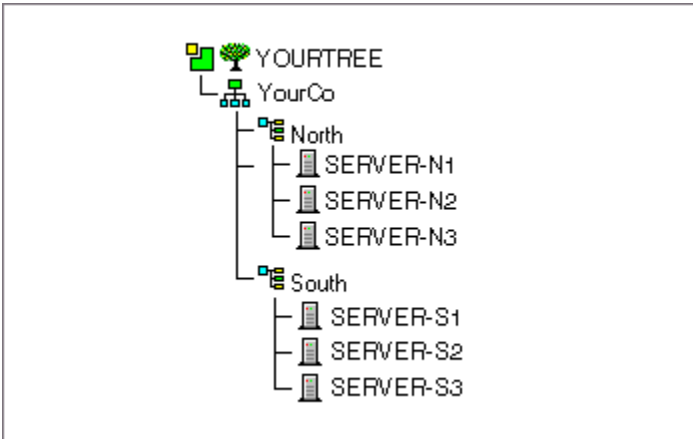
In the example above, suppose that server SERVER1 holds replicas of both the [Root] partition and the Finance partition. At this point, you haven't gained any performance advantage from NDS since SERVER1 still holds the entire Directory (replicas of both partitions).

To gain the desired performance advantage, you need to move one of the replicas to a different server. For instance, if you move the [Root] partition to SERVER2, then SERVER2 holds all objects in the [Root] and YourCo containers. SERVER1 only holds objects in the Finance and Accounts containers. The load on both SERVER1

and SERVER2 is less than it would be with no partitioning.

Partitioning and WAN Links

Suppose your network spans two sites, a North site and a South Site, separated by a WAN link. Three servers are at each site.



NDS performs faster and more reliably in this scenario if the Directory is divided in two partitions. The following explanation tells why.

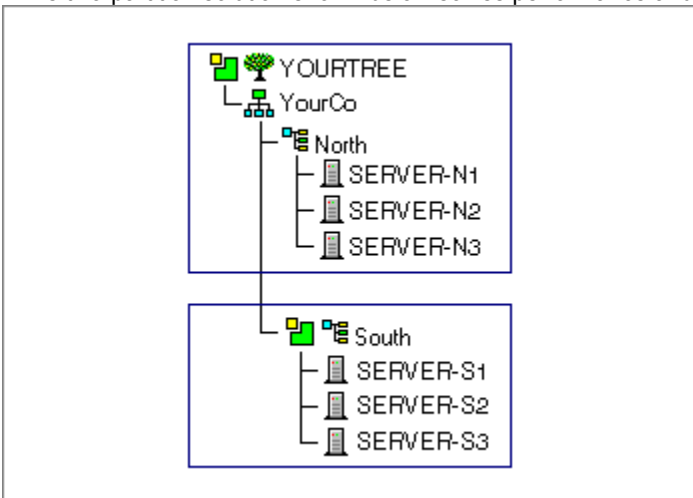
Single Partition Solution

With a single partition, the replicas are either kept at one site or distributed between the two sites. This proves unwieldy for two reasons:















- ☛ If all replicas are kept on servers at the North site, for instance, users at the South site encounter delays when logging in or accessing resources. If the link goes down, users at the South site can't log in or access resources at all.
- ☛ If replicas are distributed between sites, users can access the Directory locally. However, server-to-server synchronization of replicas happens over the WAN link, so there can be NDS errors if the link is unreliable. Any changes to the Directory are slow to propagate across the WAN link.

Two-Partition Solution

The two-partition solution shown below solves performance and reliability problems over the WAN link.



Replicas of the [Root] partition are kept on servers at the North site. Replicas of the South partition are kept on servers at the South site, as shown below.

Partition	Server	Replica Type
 [Root]	 SERVER-N1	 Master
	 SERVER-N2	 Read/write
	 SERVER-N3	 Read/write
 South	 SERVER-S1	 Master
	 SERVER-S2	 Read/write
	 SERVER-S3	 Read/write

For each site, the objects that represent local resources are kept locally. Synchronization traffic among servers also happens locally over the LAN, rather than over the slow, unreliable WAN link.

NDS traffic is generated over the WAN link, however, when a user or administrator accesses objects at a different site.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Plan Time Synchronization

NDS servers must agree on time to function properly (see [The Importance of Synchronizing Time](#)). Planning for time synchronization depends on the operating systems of servers in your network

☛ If some servers have non-NetWare* OSes (such as UNIX**), implement a third-party time synchronization product. Such products are available for Windows NT**, UNIX, and NetWare (IntranetWare*) servers.

☛ For a network of exclusively NetWare or IntranetWare servers, see [IntranetWare Time Synchronization](#).

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Plan Workgroup Containers and Leaf Objects

After designing the upper layers of the NDS* tree for an efficient partitioning strategy, you can add workgroup containers and leaf objects to the lower layers.

Create workgroup Organizational Units for sets of users that generally have common resource requirements, such as printers, file systems, and applications. If workgroups tend to have hundreds of User objects and more than one network administrator, you can subdivide workgroups around management responsibilities.

Planning an Organizational Unit for each workgroup lets you make a common trustee assignment and login script for the User objects in that workgroup. It also simplifies naming in the common context.

Workgroup Exceptions

Exceptions to general workgroup boundaries are not hard to manage. If two workgroups use a common printer, for instance, you can create an Alias to the printer in one of the workgroups. You can create Group objects to manage some User objects within a workgroup or User objects across multiple workgroups. You can create Profile objects for subsets of users with unique login script requirements.

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

Primary

A Primary time server polls and votes with other Primary time servers to determine time and provides time to Secondary time servers and client workstations. Use a Primary time server with Reference time servers to pass time to Secondary time servers and client workstations.

A Primary time server must be able to contact at least one other Primary time server or a Reference time server. It polls all other time sources to determine the correct network time and compensate for clock errors. It sets synchronization status based on its deviation from calculated network time, without regard to status of other time sources polled.

See also Time Provider Group.

Profile

Profile objects help you manage login scripts.

What a Profile Object Represents

A Profile object represents a login script which runs after the container login script and before the user login script.

Usage

Create a Profile object if you want login script commands to run for only selected users. The User objects can exist in the same container or be in different containers. Once you have created the Profile, you add the commands to its Login Script property. Then make the User objects trustees of the Profile and add the Profile to their Profile Membership property.

Important Properties

The Profile object has two important properties, as explained below: Login Script and Rights to Files and Directories.

Login Script

The login script property contains the commands you want to run for users of the Profile.

Rights to Files and Directories

If you have INCLUDE statements in the login script, you need to give the Profile object rights to the files included. This is done with the Rights to Files and Directories property. For more information on INCLUDE statements and other login script commands, see [Login Scripts](#).

Property Rights

When you make a trustee assignment, you can grant object rights and/or property rights. Object rights apply to manipulation of the entire object, while property rights only apply to certain object properties.

Property rights are shown in the lower right pane of the NWAdmin Trustees dialog below.

NWAdmin gives you two options for managing property rights:

You can manage all properties at once when the All Properties radio button is selected.

You can manage one or more individual properties when the Selected Properties radio button is selected.

Following is a description of each property right.

Supervisor allows the trustee complete power over the property.

Compare enables the trustee to compare the value of a property to a given value. This right allows searching, and only returns a true or false result. It does not allow the trustee to actually see the value of the property.

Read allows the trustee to see the values of a property. It includes the Compare right.

Write allows the trustee to create, change, and delete the values of a property.

Add Self allows the trustee to add or remove itself as a property value. It only applies to properties with object names as values, such as membership lists or Access Control Lists (ACLs).

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

Public Trustee

The [Public] trustee is not an object. It is a specialized trustee that represents any network user, logged in or not, for rights assignment purposes.

Reference

A Reference time server receives time from an external time source or from its own internal clock and provides time to Primary and Secondary time servers. Use a Reference time server when you are planning a time provider group. Do not use Reference time servers with default time synchronization.

Typically, only one Reference time server is installed on a network. If there is more than one Reference time server, each must be synchronized with the same external time source.

A Reference time server functions the same as a Primary time server except that it does not adjust its internal clock, providing a central point of time control for the entire network.

Replica Synchronization

Replica synchronization ensures that changes to Directory objects are synchronized among all replicas of the partition.

This means that any server that holds a replica of a partition must communicate with the other servers to synchronize a change.

Two types of replica synchronization can occur:

☛ Immediate sync occurs after any change to a Directory object, or any addition or deletion of an object in the NDS* tree. By default, the immediate sync process runs ten seconds after any change is saved.

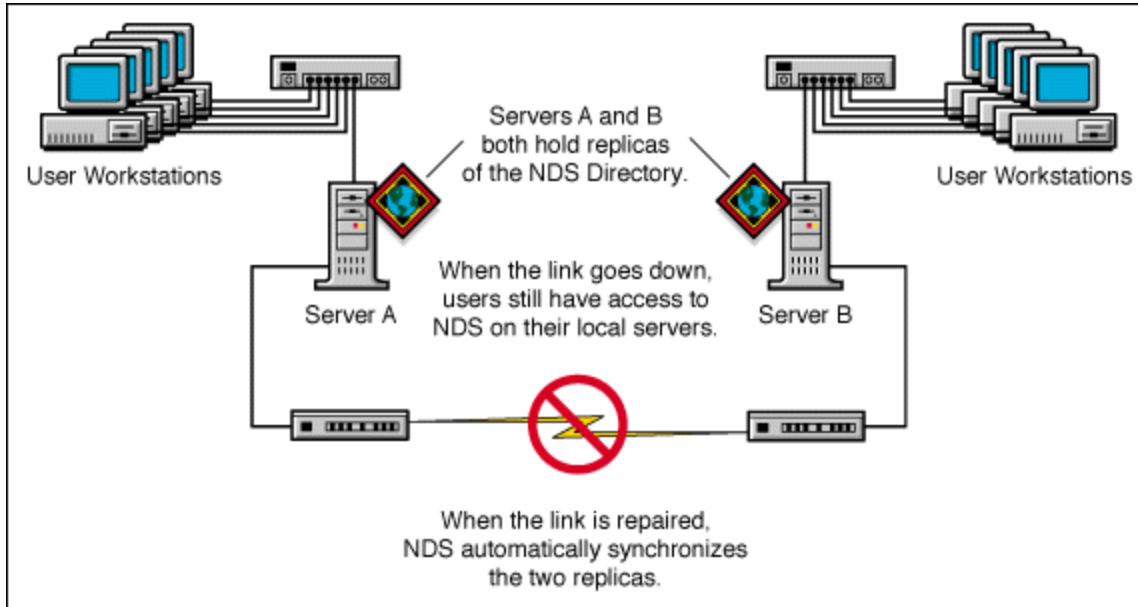
☛ Slow sync occurs for specific changes to a Directory object that are repetitive and common to multiple objects such as changes to login properties. An example of this would be a password change or an update to Login Time, Last Login Time, Network Address, and Revision attributes when a user logs in or out.

The slow sync process runs only in the absence of an immediate sync process, by default twenty-two minutes after particular changes are made.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Replicas: Keeping NDS Running

If you have more than one NDS* server on your network, you can keep multiple replicas (copies) of the NDS Directory. That way, if one server or a network link to it should fail, users can still log in and use the remaining network resources.



We recommend that you keep three replicas for fault-tolerance of NDS (assuming you have three NDS servers to store them on). A single server can hold replicas of multiple partitions.

Note: NDS replication does not provide fault tolerance for the file system. Only information about NDS objects is replicated. You can get fault tolerance for file systems by using the Transaction Tracking System* (TTS*), disk mirroring/duplexing, RAID, or Novell Replication Services* (NRS).

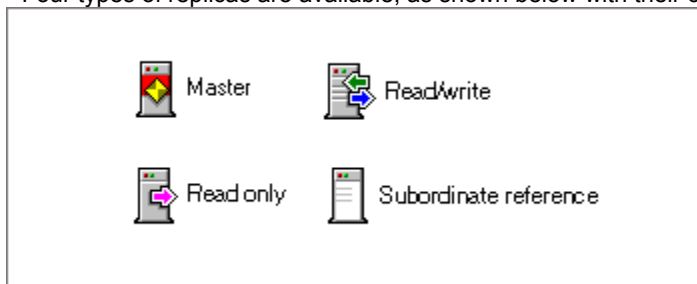
A replica is required on any server that provides bindery services, since the server requires information on all objects in its bindery context.

If users regularly access NDS information across a WAN link, you can decrease access time and WAN traffic by placing a replica containing the needed information on a server that users can access locally.

The same is true to a lesser extent on a LAN. Distributing replicas among servers on the network means information is usually retrieved from the nearest available server.

Replica Types

Four types of replicas are available, as shown below with their corresponding icons.



Master Replica

By default, the first NDS server on your network holds the master replica. There is only one master replica for each partition at a time. If other replicas are created, they are read/write replicas by default. For more information on partitions, see Partitions: Helping NDS Perform Well.

If you're going to bring down the server holding a master replica for longer than a day or two, you can make one of the read/write replicas the master. The original master replica automatically becomes read/write.

A master replica must be available on the network for NDS* to perform operations such as creating a new replica or creating a new partition.

Read/write Replica

NDS can access and change object information in a read/write replica as well as the master replica. All changes are then automatically propagated to all replicas.

If NDS responds slowly to users because of delays in the network infrastructure (such as slow WAN links or busy routers), you can create a read/write replica closer to the users who need it. You can have as many read/write replicas as you have servers to hold them, although more replicas cause more traffic to keep them synchronized with each other.

Read-only Replica

The read-only replica has been devised for future implementations of NDS. Read-only replicas receive synchronization updates from master and read/write replicas but don't receive changes directly from clients.

Subordinate Reference Replica

Subordinate reference replicas are special system-generated replicas that don't contain all the object data of a master or a read/write replica. Subordinate reference replicas therefore don't provide fault tolerance. They only contain enough information for NDS to resolve names across partition boundaries.

You can't delete a subordinate reference replica; NDS deletes it automatically when it is not needed. Subordinate reference replicas are only created on servers that hold a replica of a parent partition but no replicas of its child partitions.

If a replica of the child partition is copied to a server holding the replica of the parent, the subordinate reference replica is automatically deleted.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

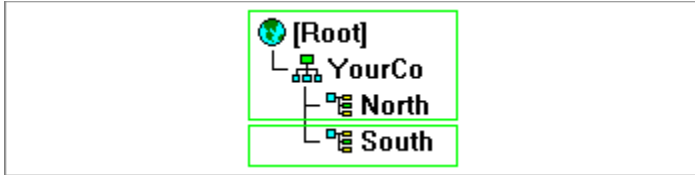
Restrict Partitions to 1,000 Objects

NDS* performs best with about 1,000 objects or fewer per partition. This number is approximate, since other variables also affect performance, such as:

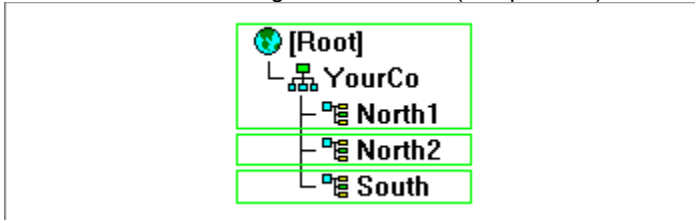
- The number and speed of servers
- The speed of network infrastructure, such as network adapters, hubs, and routers
- The amount of network traffic

Since you create partitions based on container objects, the upper levels of your NDS tree should reflect your partitioning strategy.

Suppose, for instance, that you have already planned Organizational Units (and partitions) for your company's North and South sites. Your plan for the NDS tree, with partitions shown in green, might look like the example below:



This is a good start for designing the upper levels of the tree. At this point, though, you should consider the number of objects per partition. For example, if you project 1,500 objects at the North site and 500 objects at the South site, you will want an additional Organizational Unit (and partition) at the North site. See the example below.



The additional Organizational Unit and partition should represent a distinct division of the LAN topology, such as the physical network in one building or one floor of a building.

Whenever you plan a new partition, you should also plan enough servers to hold replicas of that partition. Three replicas are recommended per partition, hosted on different NDS servers. In the example above, you would need nine NDS servers.

After designing of the upper levels of the tree, you should plan workgroup containers and leaf objects.

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

Schema Synchronization

Schema synchronization ensures that the schema is consistent across the partitions in the NDS* tree and that all schema changes are updated across the network.

This process runs once every four hours by default.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Secondary

A Secondary time server normally gets the time from a Single Reference, Primary, or Reference time server. It adjusts its internal clock to synchronize with the network time, and it provides the time to client workstations.

You can have a Secondary time server contact another Secondary time server to get the correct time. However, if the intermediate Secondary time server is unavailable, servers that contact it for the correct time might be too many hops away from a time source server to get synchronized.

A Secondary time server attempts to stay synchronized with one time source. It does not participate in voting.

Most servers should be Secondary time servers.

Security Equivalence

Security equivalence means having the same trustee assignments as another object. When you make one object security equivalent to another object, the rights of the second object are added to the rights of the first object.

For instance, suppose you make User object Joe security equivalent to the Admin object. After you create the security equivalence, Joe has the same rights to the NDS* tree as Admin.

Note: Security equivalence is only effective for one step. For instance, if you make a third user security equivalent to Joe in the example above, that user will not receive Admin rights.

Security equivalence is recorded in NDS as values in the User object's Security Equal To property.

When you add a User object as an occupant to an Organizational Role object, that User automatically becomes security equivalent to the Organizational Role object. The same is true when a User becomes a member of a Group object.

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

Single Reference

A Single Reference time server provides time to Secondary time servers and client workstations. It is typically used for smaller LANs. All servers must be able to contact the Single Reference time server.

No Primary or Reference time servers can be on the network with a Single Reference time server.

A Single Reference time server functions the same as a Reference time server, except that it does not synchronize with any other servers.

Standardize Property Values

Using standard property values such as object names makes your network more intuitive to both users and administrators. Written standards can also specify how administrators set other property values, such as telephone numbers and addresses.

Searching and browsing the Directory rely greatly on the consistency of property values. For instance, the NWAdmin Search window below shows a search for User objects with a telephone number beginning with "(801)."

If some of the values for the Telephone property were entered beginning with 1-801, the search returns an incomplete list of results.

Following is a sample document containing standards for some of the most often used properties. Of course, you only need to have standards for those properties you use. You should distribute the standards document to all administrators responsible for creating or modifying objects.

Object Class Property	Standard	Examples	Rationale
User Login name	First initial, middle initial (if applicable), last name (all lowercase), 8 characters maximum. All common names are unique in the company	msmith, bgashler	Using unique names company-wide is not required by NDS* but helps avoid conflicts within the same context (or bindery context).
User Last name	Last name (normal capitalization)	Van Pelt	Used for generation of mailing labels.
Telephone and fax numbers	Numbers separated by dashes	US: 123-456-7890 Other: 44-344-123456	Used by autodialing software.
Multiple classes Location	Two-letter location code (uppercase), dash, mail stop	BA-C23	Used by interoffice mail carriers.
Organization	YourCo for all	YourCo	If you have

Name	trees.			separate trees, a standard Organization name allows for future merging of trees.
Organizational Unit Name (based on location)	Two or three-letter location code, all uppercase	ATL, CHI, CUP, LA, BAT, BOS, DAL		
Organizational Unit Name (based on department)	Department name or abbreviation	Sales, Eng		Short, standard names are used for efficient searching.
Group Name	Descriptive name	Project Managers		
Directory Map Name	Contents of the directory indicated by the Directory Map	DOSAPPS		
Profile Name	Purpose of the profile	MobileUser		
NetWare Server Name	SERV, dash, department, dash, unique number	SERV-Eng-1		NDS requires server names to be unique in the tree

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

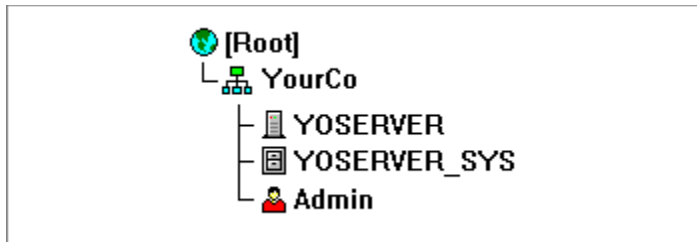
Start with the Defaults

Does your network have fewer than 1,000 objects (users and resources)? Are all servers connected by reliable LAN links? If the answer to both questions is yes, then you have a small network in NDS* terms.

For small networks, you don't need to use the more advanced features of NDS (although understanding them can help). NDS takes care of itself with its default settings. Some of these defaults are as follows:

- NDS keeps the Directory in one partition, without dividing it.
- NDS replicates the Directory to the first three NDS servers on the network. Those replicas are kept synchronized automatically.
- One NDS tree is created, rather than multiple trees.
- A single Organization object is created and named after your company.
- The Admin User object is created in the Organization container, as are the NetWare Server and Volume objects.

The picture below shows the default NDS tree for a small network. It has the required [Root] object, an Organization named after your company, a server (named YOSERVER in this example), a volume (named YOSERVER_SYS in this example), and a User object named Admin.



Note: Networks with only a Windows NT** server running NDS will have no Volume objects.

Users are added individually using NWAdmin, or in batches through a migration, upgrade, or import utility. The default NDS tree has a single container: an Organization object named after your company (YourCo in the example).

For small networks, this configuration is adequate; you can add new User objects, Application objects, printing-related objects, and other useful objects to the YourCo container.

You may want to plan new container objects in the upper levels of the tree for either or both of the reasons listed below.



Your network has slow or unreliable WAN links

Your network has more than 1,000 users

After designing of the upper levels of the tree, you should plan workgroup containers and leaf objects.

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

Synchronizing Servers in the Replica Ring

When multiple servers hold replicas of the same partition, those servers are considered a replica ring. NDS* automatically keeps those servers synchronized, so the object data is consistent on all replicas.

The following NDS processes keep servers in the replica ring synchronized.

Replica Synchronization

Schema Synchronization

Heartbeat

Limber

Backlink

Connection Management

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

The Importance of Synchronizing Time

In a single server environment, the server's internal clock is adequate to maintain a common and consistent time source for the network.

For a multiple-server network, NDS* (like many applications that use time stamps) requires that servers agree on time.

Time synchronization offers three benefits:

- ☛ Applications that run on your server provide accurate time stamps to events. Messaging and collaboration applications and databases all benefit from synchronized time.
- ☛ Workstations can be configured to get their time from servers, taking synchronized time benefits to locally-run applications.
- ☛ NDS applies correct time stamps to NDS events.

When changes are made to NDS objects, the changes can be made to different replicas on different servers. These changes must be enacted in the order in which they were requested.

An example of this would be one administrator renaming an object and another moving the object. If these events were applied out of sequence, the object might not be renamed because it would not exist in the original context.

NDS records the time of each event with a time stamp. When NDS actually modifies the database, the particular event occurs in the time and order that it was intended. NDS also uses time stamps to record time values for the network and set expiration dates.

For more information, see [Planning Time Synchronization](#).

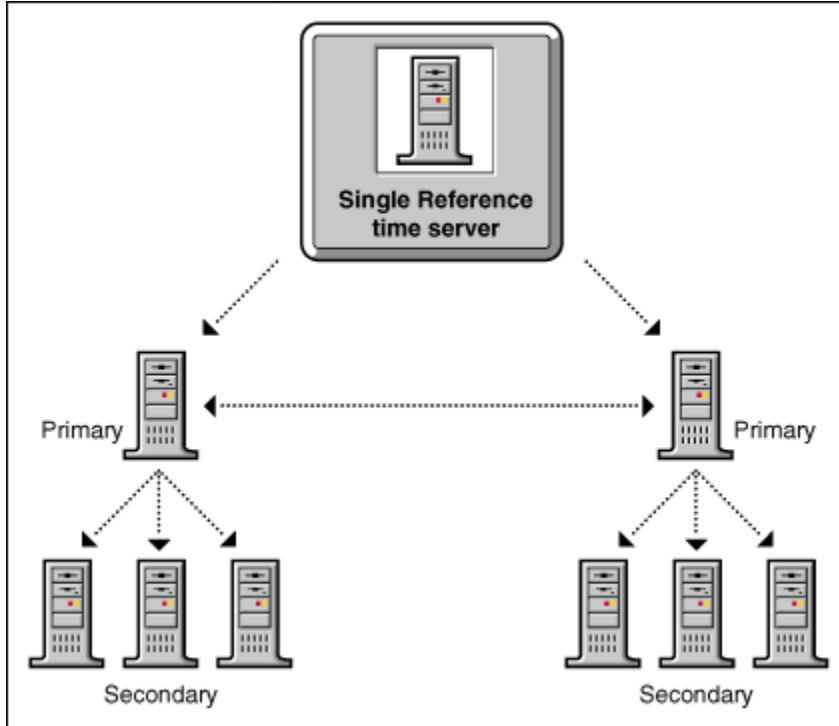
* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Time Provider Group

A time provider group consists of the following:

- One Single Reference server or one or more Reference time servers
- Two or more Primary time servers

The Single Reference (or Reference) time server polls the Primary time servers within the time provider group to vote on the correct time, as shown below.



The Single Reference (or Reference) time server provides time to Primary time servers. Primary servers, in turn, provide time to Secondary time servers. All of these servers can provide time to their own client workstations.

Reference or Single Reference?

Only one Single Reference time server can be installed in the network. One or more Reference time servers can be installed on a network. If you use more than one Reference time server, you must synchronize each Reference time server with the same external time source, such as a radio clock, atomic clock, or Internet time.

This configuration requires that you modify the time parameters on the NetWare* servers in the time provider group.

The Single Reference (or Reference) time server should be placed in a central location. Servers designated as Primary time servers should be located either at the same central location as the Reference time server or used as locational time servers at distributed sites (see below). All other servers in the network should be designated as Secondary time servers.

Time Synchronization over a WAN

If you are configuring for a multiple campus network, you should distribute Primary time servers strategically across the WAN infrastructure. This will reduce WAN traffic by providing a time source for Secondary time servers and client workstations at each location.

If your WAN infrastructure forces you to have more than seven Primary time servers in the time provider group, you should implement additional time provider groups as necessary. However, you should ensure that each Reference time server is synchronized to the same external time source. Reference time servers cannot synchronize time with other Reference time servers.

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

Time Source List

A time source list gives you more control over time synchronization than the default configuration, since you define explicitly where each server gets its time, but it requires more planning to synchronize servers efficiently.

An advantage of creating a time source list is that you maintain complete control of the time synchronization environment. Also, custom lists help eliminate nonessential network SAP traffic, as well as eliminate errors associated with accidental reconfiguration.

The time source list does require additional time for planning and installation. It is also more difficult to install or remove Primary, Reference, or Single Reference time servers. You must manually change the time source list maintained on each server.

You maintain the time source list on each server using the following parameters in the TIMESYNC.CFG file (by default in the SYS:SYSTEM directory):

```
# Configuration Parameters from server YOURSERVER

Configured Sources = ON

# Configured time source list from server YOURSERVER

Time Source = SERVER1
Time Source = SERVER2
```

With Configured Sources set to ON, the server only uses servers listed as time sources and does not listen to SAP information.

Additionally, the following parameter disables the broadcast of time information over SAP:

```
Service Advertising = OFF
```

Trademarks

Copyright © 1993-1997, Novell, Inc. All rights reserved.

Novell Trademarks

IntranetWare is a trademark of Novell, Inc.

IPX is a trademark of Novell, Inc.

NCP and NetWare Core Protocol are trademarks of Novell, Inc.

NDS is a trademark of Novell, Inc.

NetWare is a registered trademark of Novell, Inc. in the United States and other countries.

NetWare 3 is a trademark of Novell, Inc.

NetWare 4 is a trademark of Novell, Inc.

NLM is trademark of Novell, Inc.

Novell is a registered trademark of Novell, Inc. in the United States and other countries.

Novell Application Launcher is a trademark of Novell, Inc.

Novell Replication Services is a trademark of Novell, Inc.

Transaction Tracking System and TTS are trademarks of Novell, Inc.

Third-party Trademarks

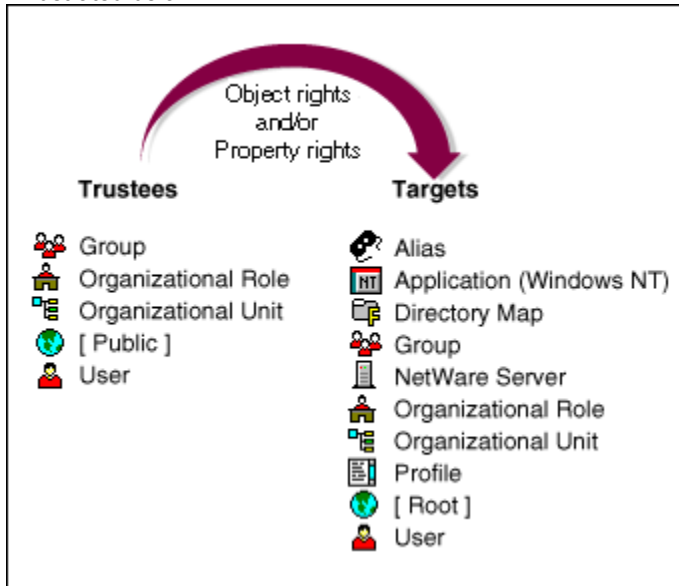
AppleTalk and Macintosh are registered trademarks of Apple Computer, Inc.

UNIX is a registered trademark of X/Open Company, Ltd., in the United States and other countries.

Windows and Windows NT are registered trademarks and Windows 95 is a trademark of Microsoft Corporation.

Trustee Assignments and Targets

The assignment of NDS* rights involves a trustee and a target object. The trustee represents the user or set of users which are receiving the authority. The target represents those network resources the users have authority over. Although any object can be either a trustee or target, objects used in typical trustee assignments are illustrated below.



Notes

☛ If you make an Alias a trustee, the rights apply only to the object it represents. The Alias object can be an explicit target, however.

☛ A file or directory in the NetWare* file system can also be a target, although file system rights are stored in the file system itself, not in NDS.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Contents

Understanding NDS

[What Is the NDS Directory?](#)

[NDS Ease of Management](#)

[Replicas: Keeping NDS Running](#)

[Context and Naming](#)

[Partitions: Helping NDS Perform Well](#)

[Object Classes and Properties](#)

[NDS Rights](#)

[Synchronizing Servers in the Replica Ring](#)

[If Your Network Needs a Bindery](#)

[The Importance of Synchronizing Time](#)

Planning Your NDS Network

[Start with the Defaults](#)

[Accommodate Slow or Unreliable WAN Links](#)

[Restrict Partitions to 1,000 Objects](#)

[Plan Workgroup Containers and Leaf Objects](#)

[Standardize Property Values](#)

[Plan Time Synchronization](#)

User

A User object is required for logging in. When you install the first server into an NDS* tree, a User object named Admin is created. The first time you log in, you will use the Admin account.


You can create User objects with NWAdmin, or import them automatically from Windows NT** domains using NDS for NT. You can also import User objects in batches from database files. Additionally, NetWare upgrade utilities import Users from existing bindery servers.

What a User Object Represents


A User object represents a person who uses the network.


Usage

You should create User objects for all users who need to use the network. Although you can manage User objects individually, you'll save time by

-  Using Template objects to set default properties for most User objects. The Template applies automatically to new Users you create (not to already-existing ones).

-  Creating Group objects to manage sets of Users.

-  Assigning rights using the container objects as trustees when you want that assignment to apply to all User objects in the container.

-  Selecting multiple Users. When you do (with standard Windows** shift-clicking or control-clicking), you can change property values for all selected User objects.

Important Properties

User objects have over 80 properties. For a complete list of properties, select a User object in NWAdmin, then choose Object > Details. To display a description for each page of properties, click Help.

The Login Name and Last Name properties are required. These and some of the most useful properties are listed below.

Account Expiration Date

This property lets you limit the life of a user account. After the expiration date, the account is locked so the user cannot log in.

Account Disabled

This property has a system-generated value that indicates a lock on the account so the user cannot log in. The lock might occur if the account has expired or because the user has given too many incorrect passwords in succession.

Force Periodic Password Changes

This property allows you to enhance security by requiring the user to change passwords after a specified interval.

Group Membership

This property lists all the Group Objects that include the User as a member.

Home Directory

The Home Directory property refers to a NetWare volume and file system path for the user's own files. Most administrators like to create such a directory so that a user's working files can be kept on the network.

The directory referred to in this property can be created automatically when you create the User object.

Last Login

This is a system-generated property that lists the date and time that the user last logged in.

Last Name

The Last Name property, though required, is not used directly by NDS. Applications that take advantage of the NDS name base can use this property, along with other identification properties such as Given Name, Title, Location, and Fax Number.

Limit Concurrent Connections

This property allows you to set the maximum number of sessions a user can have on the network at any given time.

Login Name

This is the name shown in NWAdmin by the User icon. It is also the name supplied by the user when logging in. NDS does not require that login names be unique throughout the network, only in each container. However, you may want to keep login names unique across the company to simplify administration.

Typically, login names are a combination of first and last names, such as JOHNT or JTHOMAS for John Thomas.

Login Script

The Login Script property allows you to create specific login commands for a User. When a user logs in, the container login script runs first. Then a profile login script runs if the User has been added to the membership list of a Profile object. Finally, the user login script runs (if one exists).

You should put most of the login commands in container login scripts to save administrative time. The User login script can be edited to manage unique exceptions to common needs.

Login Time Restrictions

This property allows you to set times and days when the user can log in.

Network Addresses

This property contains system-generated values that list all the IPX and/or IP addresses from which the User is logged in. These values are useful for troubleshooting network problems at the packet level.

Require a Password

This property allows you to control whether the user must use a password. Other related properties let you set common password constraints such as password length.

Rights to Files and Directories

This property lists all rights assignments made for this user to the NetWare* file system. Using NWAdmin, you can also check a user's effective rights to files and directories, which include those inherited from other objects.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

Volume

When you create a physical volume on a server, a Volume object is automatically created in the NDS* tree. By default, the name of the Volume object is the server's name with an underscore and the physical volume's name appended (example: YOSERVER_SYS).

What a Volume Object Represents

A Volume object represents a physical volume on a server, whether it is a writable disk, a CD-ROM, or other storage medium. The Volume object in NDS does not contain information about the files and directories on that volume, though you can access that information through NWAdmin. File and directory information is retained in the file system itself.

Usage

Using NWAdmin, you can click the Volume icon to [manage files and directories](#) on that volume. NWAdmin provides information about the volume's free disk space, directory entry space, and compression statistics.

You can also [create Volume objects](#) in the NDS tree for NetWare 2 and NetWare 3 volumes.

Important Properties

In addition to the required Name and Host Volume properties, there are other important Volume properties.

Name

This is the name of the Volume object in the NDS tree. By default, this name derives from the name of the physical volume, though you can change the object name.

Host Server

This is the server on which the volume resides.

Version

The Version property gives the NetWare or NDS version of the server hosting the volume.

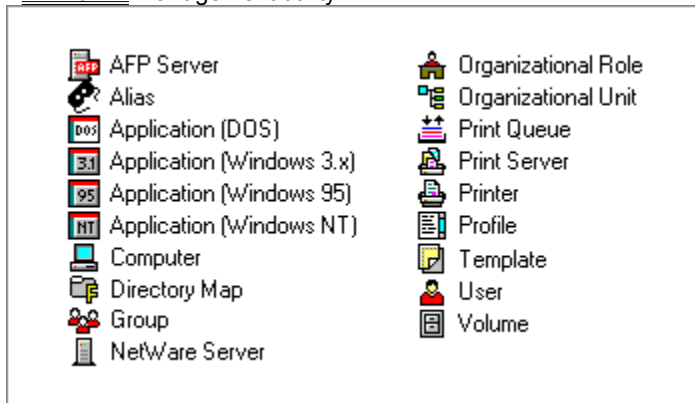
Host Volume

This is the physical volume name. Since the actual Volume object name does not need to reflect the physical volume name, this property is necessary to associate the Volume object with the physical volume.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

What Is the NDS Directory?

In simplest terms, the NDS* Directory is a list of objects that represent network resources, such as network users, servers, printers, print queues, and applications. The figure below shows a few of the objects as viewed in the [NWAdmin](#) management utility.



Note: Some object classes may not be available, depending on the operating system of the [NDS server](#). For more information on NDS objects, see [Object Classes and Properties](#).

The NDS Directory is physically stored as a set of database files on a server. If the server hosts file system volumes, these files are on volume SYS. If no volumes are present, the Directory is stored on the server's local drive in a subdirectory of the NDS installation directory.

If you have more than one NDS server on the network, the NDS Directory can be [replicated](#) on multiple servers.

* Novell trademark. ** Third-party trademark. For more information, see [Trademarks](#).

[Root]

The [Root] container is created when you first install NDS* on a server in your network. As the topmost container, it usually holds Organization objects, Country objects, or Alias objects.

What [Root] Represents

[Root] represents the beginning of your NDS tree.

Usage

[Root] is used to make universal rights assignments. Because of inheritance, any rights assignments you make to [Root] as the target apply to all objects in the tree (see also NDS Rights). The [Public] trustee has the Browse right and Admin has the Supervisor right to [Root] by default.

Important Properties

The [Root] object has a Name property, which is the tree name you supplied when installing the first server. The tree name is shown in the title bar of NWAdmin.

* Novell trademark. ** Third-party trademark. For more information, see Trademarks.

